
django-centralauth

unknown

May 12, 2021

CONTENTS

1	Features	3
2	Requirements	5
3	Prepare for development	7
4	Resources	9
4.1	Installation	9
4.2	Changelog	11
4.3	API Reference	12
5	Indices and tables	25
	Python Module Index	27
	Index	29

django-centralauth solves the problem of managing user access and permissions from multiple projects in one central place.

FEATURES

- based on OAuth2 standard.
- provider app to set up your own user-management application.
- client app for delegating authentication and permissions management to provider.

REQUIREMENTS

django-centralauth supports Python 3 only and requires at least Django 2. and django-oauth-toolkit.

PREPARE FOR DEVELOPMENT

```
$ poetry install
```

Now you're ready to run the tests:

```
$ poetry run py.test
```


RESOURCES

- [Documentation](#)
- [Bug Tracker](#)
- [Code](#)

Contents:

4.1 Installation

- Install with pip:

```
pip install django-centralauth
```

4.1.1 Provider side

You need to update some of your Django settings.

- Your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (  
    # ...  
    'oauth2_provider',  
    'centralauth.provider',  
)
```

- Your `MIDDLEWARE` setting:

```
MIDDLEWARE = [  
    'oauth2_provider.middleware.OAuth2TokenMiddleware',  
    # ...  
)
```

- Your `AUTHENTICATION_BACKENDS` setting:

```
AUTHENTICATION_BACKENDS = (  
    'oauth2_provider.backends.OAuth2Backend',  
    # ...  
)
```

- Add the following settings in addition:

```
OAUTH2_PROVIDER_ACCESS_TOKEN_MODEL = 'oauth2_provider.AccessToken'  
OAUTH2_PROVIDER_APPLICATION_MODEL = 'provider.Application'
```

- Configure the OAuth2 provider backend class:

```
OAUTH2_PROVIDER = {  
    'OAUTH2_BACKEND_CLASS': 'centralauth.provider.oauth2_backends.  
↪CentralauthOAuthBackend',  
}
```

If you want to re-validate the access more often, you might reduce the lifetime of the generated access tokens:

```
OAUTH2_PROVIDER = {  
    # ...  
    'ACCESS_TOKEN_EXPIRE_SECONDS': 5 * 60,  
}
```

After you updated your settings, add the `centralauth.provider` urls to your url configuration:

```
urlpatterns = [  
    # ...  
    path('provider/', include('centralauth.provider.urls'))  
]
```

Note: Make sure that you configure a sane `LOGIN_URL`. `django-oauth-toolkit` will redirect users to this url to ensure the requesting user is logged in.

4.1.2 Client side

You need to update some of your Django settings.

- Your `INSTALLED_APPS` setting:

```
INSTALLED_APPS = (  
    # ...  
    'centralauth.client',  
)
```

- Your `AUTHENTICATION_BACKENDS` setting:

```
# Disable regular logins using local users and enforce centralauth logins.  
AUTHENTICATION_BACKENDS = (  
    'centralauth.client.backends.OAuthBackend'  
)
```

- Add the following settings in addition:

```
# The full uri to the provider side urls.  
CENTRALAUTH_PROVIDER_URL = 'http://localhost:8000/provider'  
  
# The application credentials generated on the provider side using the Django admin.
```

(continues on next page)

(continued from previous page)

```
CENTRALAUTH_CLIENT_ID = 'ADD-YOUR-CLIENT-ID'  
CENTRALAUTH_CLIENT_SECRET = 'ADD-YOUR-CLIENT-SECRET'
```

After you updated your settings, add the `centralauth.client` urls to your url configuration:

```
urlpatterns = [  
    # ...  
    path('centralauth/', include('centralauth.client.urls'))  
]
```

Note: Centralauth provides an option to hijack the admin login interface to make sure that the users go through the Centralauth oauth login flow.

You might set `CENTRALAUTH_CUSTOM_LOGIN_TEMPLATE` to `True` or provide a Django template path to your custom template.

4.2 Changelog

4.2.1 2.0.0 (2021-05-04)

- Add support for Python 3.7, 3.8 and 3.9
- Add support for Django 3
- Drop support for Python 2
- Drop support for Django older than 2.2

4.2.2 1.2.0 (2019-02-22)

- django-centralauth now depends on requests-oauthlib \geq 1.2.0 and therefore oauthlib \geq 3.0
- Fixed some race conditions in middleware (when tokens are refreshed twice)

4.2.3 1.1.2 (2019-02-12)

- Fix bug in `get_or_create` of permission sync api endpoint

4.2.4 1.1.1 (2019-01-10)

- Fix permissions sync - remove deleted permissions

4.2.5 1.1.0 (2018-11-26)

- Improve permission updates on user sync (don't use clear, just merge source and target set)

4.2.6 1.0.0 (2018-11-22)

- Initial release of *django-centralauth*

Api documentation:

4.3 API Reference

4.3.1 centralauth package

Subpackages

centralauth.client package

Submodules

centralauth.client.apps module

```
class centralauth.client.apps.ClientConfig(app_name, app_module)  
    Bases: django.apps.config.AppConfig  
    name = 'centralauth.client'  
    ready()  
        Override this method in subclasses to run code when Django starts.
```

centralauth.client.backends module

```
class centralauth.client.backends.OAuthBackend  
    Bases: django.contrib.auth.backends.ModelBackend  
    Authenticate user with OAuth access token.  
    authenticate(request, token)  
    get_user(user_id)
```

centralauth.client.constants module

```
centralauth.client.constants.provider_url()
```


centralauth.client.middleware module

```
class centralauth.client.middleware.CentralAuthSyncMiddleware(get_response=None)
    Bases: django.utils.deprecation.MiddlewareMixin
    process_request(request, retries=0)
```

centralauth.client.services module

```
centralauth.client.services.serialize_perm(perm)
    Serialize given permission object.
```

Returns keys: app_label, codename, repr.

Return type dict

```
centralauth.client.services.register_perms()
    Register permissions available in the project to the provider.
```

```
centralauth.client.services.oauth2_client(token, session=None)
```

```
centralauth.client.services.save_token(session, token)
```

```
centralauth.client.services.load_token(session)
```

```
centralauth.client.services.sync_user(user, client)
```

```
centralauth.client.services.user_details(client)
```

```
centralauth.client.services.update_user(user, **kwargs)
    Update user attributes and permissions
```

centralauth.client.urls module**centralauth.client.views module**

```
centralauth.client.views.get_oauth_redirect_url(request, next_url=None)
```

```
class centralauth.client.views.LoginView(**kwargs)
    Bases: django.views.generic.base.View
```

Request authorization code from provider.

Set next url in session. Build authorization code request and redirect to provider server.

```
get(*args, **kwargs)
```

```
class centralauth.client.views.CallbackView(**kwargs)
    Bases: django.views.generic.base.View
```

Exchange authorization code for access token and authenticate user.

Authenticate user with access token. Redirect to next_url or admin page.

```
get(*args, **kwargs)
```

centralauth.provider package**Submodules****centralauth.provider.admin module**

```
class centralauth.provider.admin.ApplicationPermissionGroupAdminForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    class Meta
        Bases: object

        widgets = {'permissions': <django.contrib.admin.widgets.FilteredSelectMultiple
            object>}

    base_fields = {'users': <django.forms.models.ModelMultipleChoiceField object>}
    declared_fields = {'users': <django.forms.models.ModelMultipleChoiceField object>}

    property media
        Return all media required to render the widgets on this form.

class centralauth.provider.admin.ApplicationPermissionGroupAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    list_display = ('name', 'application')
    list_filter = ('application',)
    search_fields = ('name', 'applicationuser__user__username')

    form
        alias of centralauth.provider.admin.ApplicationPermissionGroupAdminForm

    fields = ('name', 'application')

    get_fields(request, obj=None)
        Hook for specifying fields.

    get_readonly_fields(request, obj=None)
        Hook for specifying custom readonly fields.

    get_inline_instances(request, obj=None)

    save_related(request, form, *args, **kwargs)
        Given the HttpRequest, the parent ModelForm instance, the list of inline formsets and a boolean value
        based on whether the parent is being added or changed, save the related objects to the database. Note that
        at this point save_form() and save_model() have already been called.

    property media

class centralauth.provider.admin.ApplicationUserAdminForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    base_fields = {}
    declared_fields = {}

    property media
        Return all media required to render the widgets on this form.

class centralauth.provider.admin.ApplicationUserAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
```

```

list_display = ('user', 'application')
list_filter = ('application',)
search_fields = ('user__username',)

form
    alias of centralauth.provider.admin.ApplicationUserAdminForm

fields = ('user', 'application', 'is_superuser', 'is_staff', 'is_active')

get_fields(request, obj=None)
    Hook for specifying fields.

get_readonly_fields(request, obj=None)
    Hook for specifying custom readonly fields.

property media

class centralauth.provider.admin.ApplicationUserInlineFormset(data=None, files=None,
                                                            instance=None,
                                                            save_as_new=False, prefix=None,
                                                            queryset=None, **kwargs)

Bases: django.forms.models.BaseInlineFormSet

get_form_kwargs(index)
    Return additional keyword arguments for each individual formset form.

    index will be None if the form being constructed is a new empty form.

class centralauth.provider.admin.ApplicationUserInline(*args, **kwargs)
Bases: django.forms.models.ModelForm

base_fields = {}

declared_fields = {}

property media
    Return all media required to render the widgets on this form.

class centralauth.provider.admin.ApplicationUserInline(parent_model, admin_site)
Bases: django.contrib.admin.options.TabularInline

model
    alias of centralauth.provider.models.ApplicationUser

formset
    alias of centralauth.provider.admin.ApplicationUserInlineFormset

form
    alias of centralauth.provider.admin.ApplicationUserInlineForm

exclude = ('permissions',)

extra = 0

property media

class centralauth.provider.admin.ApplicationAdminForm(*args, **kwargs)
Bases: django.forms.models.ModelForm

base_fields = {}

declared_fields = {}

```

property media

Return all media required to render the widgets on this form.

```
class centralauth.provider.admin.ApplicationAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    list_display = ('name', 'redirect_uris')

    form
        alias of centralauth.provider.admin.ApplicationAdminForm

    fields = ('name', 'client_id', 'client_secret', 'redirect_uris')

    inlines = [<class 'centralauth.provider.admin.ApplicationUserInline'>]

    get_inline_instances(request, obj=None)

    property media
```

centralauth.provider.apps module

```
class centralauth.provider.apps.ProviderConfig(app_name, app_module)
    Bases: django.apps.config.AppConfig

    name = 'centralauth.provider'
```

centralauth.provider.models module

```
class centralauth.provider.models.Application(*args, **kwargs)
    Bases: oauth2_provider.models.AbstractApplication

    Centralauth custom application model.

    save(*args, **kwargs)
        Save the current instance. Override this in a subclass if you want to control the saving process.

        The 'force_insert' and 'force_update' parameters can be used to insist that the "save" must be an SQL insert
        or update (or equivalent for non-SQL backends), respectively. Normally, they should not be set.

    exception DoesNotExist
        Bases: django.core.exceptions.ObjectDoesNotExist

    exception MultipleObjectsReturned
        Bases: django.core.exceptions.MultipleObjectsReturned

    accesstoken_set
        Accessor to the related objects manager on the reverse side of a many-to-one relation.
```

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

applicationpermission_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

applicationpermissiongroup_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

applicationuser_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

get_algorithm_display(* , field=<django.db.models.fields.CharField: algorithm>)

get_authorization_grant_type_display(* , field=<django.db.models.fields.CharField: authorization_grant_type>)

get_client_type_display(* , field=<django.db.models.fields.CharField: client_type>)

get_next_by_created(* , field=<django.db.models.fields.DateTimeField: created>, is_next=True, **kwargs)

get_next_by_updated(* , field=<django.db.models.fields.DateTimeField: updated>, is_next=True, **kwargs)

get_previous_by_created(* , field=<django.db.models.fields.DateTimeField: created>, is_next=False, **kwargs)

get_previous_by_updated(* , field=<django.db.models.fields.DateTimeField: updated>, is_next=False, **kwargs)

grant_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

idtoken_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

objects = <django.db.models.manager.Manager object>

refresh_token_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

```
class centralauth.provider.models.ApplicationPermission(*args, **kwargs)
```

Bases: django.db.models.base.Model

Model for holding all permissions available for application.

application

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

repr

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

codename

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

app_label

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

date_created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

application_id

applicationpermissiongroup_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

applicationuser_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

```
get_next_by_date_created(*, field=<django.db.models.fields.DateTimeField: date_created>,
                        is_next=True, **kwargs)
```

```
get_previous_by_date_created(*, field=<django.db.models.fields.DateTimeField: date_created>,
                             is_next=False, **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`objects = <django.db.models.manager.Manager object>`

class `centralauth.provider.models.ApplicationPermissionGroup(*args, **kwargs)`

Bases: `django.db.models.base.Model`

Model for for managing groups of permissions.

Permission groups are not synced with client Group objects. In client all permissions are handled on Permission object level.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

application

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

permissions

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

date_created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

application_id

applicationuser_set

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_next_by_date_created(**field*=<`django.db.models.fields.DateTimeField: date_created`>, *is_next*=`True`, ***kwargs*)


```
get_previous_by_date_created(*, field=<django.db.models.fields.DateTimeField: date_created>,
                             is_next=False, **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

```
class centralauth.provider.models.ApplicationUser(*args, **kwargs)
```

Bases: django.db.models.base.Model

Model for managing user permissions within application.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToManyDescriptor instance.

application

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToManyDescriptor instance.

is_superuser

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_staff

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_active

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

permissions

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

date_created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_permissions()

Combine all user permissions.

Returns list of ids of all user permissions.

Return type list

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

application_id

`get_next_by_date_created(*, field=<django.db.models.fields.DateTimeField: date_created>, is_next=True, **kwargs)`

`get_previous_by_date_created(*, field=<django.db.models.fields.DateTimeField: date_created>, is_next=False, **kwargs)`

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`objects = <django.db.models.manager.Manager object>`

`user_id`

centralauth.provider.oauth_urls module

centralauth.provider.urls module

centralauth.provider.views module

`class centralauth.provider.views.UserEndpoint(**kwargs)`

Bases: `oauth2_provider.views.generic.ProtectedResourceView`

Endpoint for accessing user data.

This endpoint can be accessed only with valid access token in header.

`get(request, *args, **kwargs)`

```
class centralauth.provider.views.PermEndpoint(**kwargs)
    Bases: django.views.generic.base.View
    dispatch(request, *args, **kwargs)
    post(request, *args, **kwargs)
```


INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

- centralauth, 12
- centralauth.client, 12
 - centralauth.client.apps, 12
 - centralauth.client.backends, 12
 - centralauth.client.constants, 12
 - centralauth.client.middleware, 13
 - centralauth.client.services, 13
 - centralauth.client.urls, 13
 - centralauth.client.views, 13
- centralauth.provider, 14
 - centralauth.provider.admin, 14
 - centralauth.provider.apps, 16
 - centralauth.provider.models, 16
 - centralauth.provider.oauth_urls, 22
 - centralauth.provider.urls, 22
 - centralauth.provider.views, 22

INDEX

A

- `accesstoken_set` (*centralauth.provider.models.Application* attribute), 16
- `app_label` (*centralauth.provider.models.ApplicationPermission* attribute), 19
- `application` (*centralauth.provider.models.ApplicationPermission* attribute), 18
- `application` (*centralauth.provider.models.ApplicationPermissionGroup* attribute), 20
- `application` (*centralauth.provider.models.ApplicationUser* attribute), 21
- `Application` (class in *centralauth.provider.models*), 16
- `Application.DoesNotExist`, 16
- `Application.MultipleObjectsReturned`, 16
- `application_id` (*centralauth.provider.models.ApplicationPermission* attribute), 19
- `application_id` (*centralauth.provider.models.ApplicationPermissionGroup* attribute), 20
- `application_id` (*centralauth.provider.models.ApplicationUser* attribute), 22
- `ApplicationAdmin` (class in *centralauth.provider.admin*), 16
- `ApplicationAdminForm` (class in *centralauth.provider.admin*), 15
- `ApplicationPermission` (class in *centralauth.provider.models*), 18
- `ApplicationPermission.DoesNotExist`, 19
- `ApplicationPermission.MultipleObjectsReturned`, 19
- `applicationpermission_set` (*centralauth.provider.models.Application* attribute), 16
- `ApplicationPermissionGroup` (class in *centralauth.provider.models*), 20
- `ApplicationPermissionGroup.DoesNotExist`, 20
- `ApplicationPermissionGroup.MultipleObjectsReturned`, 20
- `applicationpermissiongroup_set` (*centralauth.provider.models.Application* attribute), 17
- `applicationpermissiongroup_set` (*centralauth.provider.models.ApplicationPermission* attribute), 19
- `ApplicationPermissionGroupAdmin` (class in *centralauth.provider.admin*), 14
- `ApplicationPermissionGroupAdminForm` (class in *centralauth.provider.admin*), 14
- `ApplicationPermissionGroupAdminForm.Meta` (class in *centralauth.provider.admin*), 14
- `ApplicationUser` (class in *centralauth.provider.models*), 21
- `ApplicationUser.DoesNotExist`, 22
- `ApplicationUser.MultipleObjectsReturned`, 22
- `applicationuser_set` (*centralauth.provider.models.Application* attribute), 17
- `applicationuser_set` (*centralauth.provider.models.ApplicationPermission* attribute), 19
- `applicationuser_set` (*centralauth.provider.models.ApplicationPermissionGroup* attribute), 20
- `ApplicationUserAdmin` (class in *centralauth.provider.admin*), 14
- `ApplicationUserAdminForm` (class in *centralauth.provider.admin*), 14
- `ApplicationUserInline` (class in *centralauth.provider.admin*), 15
- `ApplicationUserInlineForm` (class in *centralauth.provider.admin*), 15
- `ApplicationUserInlineFormset` (class in *centralauth.provider.admin*), 15
- `authenticate()` (*centralauth.client.backends.OAuthBackend* method), 12

B

- `base_fields` (*centralauth.provider.admin.ApplicationAdminForm* attribute), 15
- `base_fields` (*centralauth.provider.admin.ApplicationPermissionGroupAdminForm* attribute), 15

- attribute*), 14
 - `base_fields` (*centralauth.provider.admin.ApplicationUserAdminForm* *attribute*), 14
 - `base_fields` (*centralauth.provider.admin.ApplicationUserAdminForm* *attribute*), 15
- C**
- `CallbackView` (*class in centralauth.client.views*), 13
 - `centralauth`
 - module, 12
 - `centralauth.client`
 - module, 12
 - `centralauth.client.apps`
 - module, 12
 - `centralauth.client.backends`
 - module, 12
 - `centralauth.client.constants`
 - module, 12
 - `centralauth.client.middleware`
 - module, 13
 - `centralauth.client.services`
 - module, 13
 - `centralauth.client.urls`
 - module, 13
 - `centralauth.client.views`
 - module, 13
 - `centralauth.provider`
 - module, 14
 - `centralauth.provider.admin`
 - module, 14
 - `centralauth.provider.apps`
 - module, 16
 - `centralauth.provider.models`
 - module, 16
 - `centralauth.provider.oauth_urls`
 - module, 22
 - `centralauth.provider.urls`
 - module, 22
 - `centralauth.provider.views`
 - module, 22
 - `CentralAuthSyncMiddleware` (*class in centralauth.client.middleware*), 13
 - `ClientConfig` (*class in centralauth.client.apps*), 12
 - `codename` (*centralauth.provider.models.ApplicationPermission* *attribute*), 19
- D**
- `date_created` (*centralauth.provider.models.ApplicationPermission* *attribute*), 19
 - `date_created` (*centralauth.provider.models.ApplicationPermissionGroup* *attribute*), 20
 - `date_created` (*centralauth.provider.models.ApplicationUser* *attribute*), 22
- `declared_fields` (*centralauth.provider.admin.ApplicationAdminForm* *attribute*), 15
 - `declared_fields` (*centralauth.provider.admin.ApplicationPermissionGroupAdminForm* *attribute*), 14
 - `declared_fields` (*centralauth.provider.admin.ApplicationUserAdminForm* *attribute*), 14
 - `declared_fields` (*centralauth.provider.admin.ApplicationUserInlineForm* *attribute*), 15
 - `dispatch()` (*centralauth.provider.views.PermEndpoint* *method*), 23
- E**
- `exclude` (*centralauth.provider.admin.ApplicationUserInline* *attribute*), 15
 - `extra` (*centralauth.provider.admin.ApplicationUserInline* *attribute*), 15
- F**
- `fields` (*centralauth.provider.admin.ApplicationAdmin* *attribute*), 16
 - `fields` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* *attribute*), 14
 - `fields` (*centralauth.provider.admin.ApplicationUserAdmin* *attribute*), 15
 - `form` (*centralauth.provider.admin.ApplicationAdmin* *attribute*), 16
 - `form` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* *attribute*), 14
 - `form` (*centralauth.provider.admin.ApplicationUserAdmin* *attribute*), 15
 - `form` (*centralauth.provider.admin.ApplicationUserInline* *attribute*), 15
 - `formset` (*centralauth.provider.admin.ApplicationUserInline* *attribute*), 15
- G**
- `get()` (*centralauth.client.views.CallbackView* *method*), 13
 - `get()` (*centralauth.client.views.LoginView* *method*), 13
 - `get()` (*centralauth.provider.views.UserEndpoint* *method*), 22
 - `get_algorithm_display()` (*centralauth.provider.models.Application* *method*), 17
 - `get_authorization_grant_type_display()` (*centralauth.provider.models.Application* *method*), 17
 - `get_client_type_display()` (*centralauth.provider.models.Application* *method*), 17

`get_fields()` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* method), 14
`get_fields()` (*centralauth.provider.admin.ApplicationUserAdmin* method), 15
`get_form_kwargs()` (*centralauth.provider.admin.ApplicationUserInlineFormset* method), 15
`get_inline_instances()` (*centralauth.provider.admin.ApplicationAdmin* method), 16
`get_inline_instances()` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* method), 14
`get_next_by_created()` (*centralauth.provider.models.Application* method), 17
`get_next_by_date_created()` (*centralauth.provider.models.ApplicationPermission* method), 19
`get_next_by_date_created()` (*centralauth.provider.models.ApplicationPermissionGroup* method), 20
`get_next_by_date_created()` (*centralauth.provider.models.ApplicationUser* method), 22
`get_next_by_updated()` (*centralauth.provider.models.Application* method), 17
`get_oauth_redirect_url()` (in module *centralauth.client.views*), 13
`get_permissions()` (*centralauth.provider.models.ApplicationUser* method), 22
`get_previous_by_created()` (*centralauth.provider.models.Application* method), 17
`get_previous_by_date_created()` (*centralauth.provider.models.ApplicationPermission* method), 19
`get_previous_by_date_created()` (*centralauth.provider.models.ApplicationPermissionGroup* method), 21
`get_previous_by_date_created()` (*centralauth.provider.models.ApplicationUser* method), 22
`get_previous_by_updated()` (*centralauth.provider.models.Application* method), 17
`get_readonly_fields()` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* method), 14
`get_readonly_fields()` (*centralauth.provider.admin.ApplicationUserAdmin* method), 15
`get_token_set()` (*centralauth.provider.models.ApplicationPermission* attribute), 19
`get_token_set()` (*centralauth.provider.models.ApplicationPermissionGroup* attribute), 21
`get_token_set()` (*centralauth.provider.models.ApplicationUser* attribute), 22
`idtoken_set` (*centralauth.provider.models.Application* attribute), 18
`inlines` (*centralauth.provider.admin.ApplicationAdmin* attribute), 16
`is_active` (*centralauth.provider.models.ApplicationUser* attribute), 21
`is_staff` (*centralauth.provider.models.ApplicationUser* attribute), 21
`is_superuser` (*centralauth.provider.models.ApplicationUser* attribute), 21
L
`list_display` (*centralauth.provider.admin.ApplicationAdmin* attribute), 16
`list_display` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* attribute), 14
`list_display` (*centralauth.provider.admin.ApplicationUserAdmin* attribute), 14
`list_filter` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* attribute), 14
`list_filter` (*centralauth.provider.admin.ApplicationUserAdmin* attribute), 15
`load_token()` (in module *centralauth.client.services*), 13
`LoginView` (class in *centralauth.client.views*), 13
M
`media` (*centralauth.provider.admin.ApplicationAdmin* property), 16
`media` (*centralauth.provider.admin.ApplicationAdminForm* property), 15
`media` (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* property), 14
`media` (*centralauth.provider.admin.ApplicationPermissionGroupAdminForm* property), 14
`media` (*centralauth.provider.admin.ApplicationUserAdmin* property), 15
`media` (*centralauth.provider.admin.ApplicationUserAdminForm* property), 14

- media (*centralauth.provider.admin.ApplicationUserInline* property), 15
- media (*centralauth.provider.admin.ApplicationUserInlineForm* property), 15
- model (*centralauth.provider.admin.ApplicationUserInline* attribute), 15
- module
- centralauth, 12
 - centralauth.client, 12
 - centralauth.client.apps, 12
 - centralauth.client.backends, 12
 - centralauth.client.constants, 12
 - centralauth.client.middleware, 13
 - centralauth.client.services, 13
 - centralauth.client.urls, 13
 - centralauth.client.views, 13
 - centralauth.provider, 14
 - centralauth.provider.admin, 14
 - centralauth.provider.apps, 16
 - centralauth.provider.models, 16
 - centralauth.provider.oauth_urls, 22
 - centralauth.provider.urls, 22
 - centralauth.provider.views, 22
- N**
- name (*centralauth.client.apps.ClientConfig* attribute), 12
- name (*centralauth.provider.apps.ProviderConfig* attribute), 16
- name (*centralauth.provider.models.ApplicationPermissionGroup* attribute), 20
- O**
- oauth2_client() (in module *centralauth.client.services*), 13
- OAuthBackend (class in *centralauth.client.backends*), 12
- objects (*centralauth.provider.models.Application* attribute), 18
- objects (*centralauth.provider.models.ApplicationPermission* attribute), 19
- objects (*centralauth.provider.models.ApplicationPermissionGroup* attribute), 21
- objects (*centralauth.provider.models.ApplicationUser* attribute), 22
- P**
- permissions (*centralauth.provider.models.ApplicationPermissionGroup* attribute), 20
- permissions (*centralauth.provider.models.ApplicationUser* attribute), 21
- PermsEndpoint (class in *centralauth.provider.views*), 22
- post() (*centralauth.provider.views.PermsEndpoint* method), 23
- process_request() (*centralauth.client.middleware.CentralAuthSyncMiddleware* method), 13
- provider_url() (in module *centralauth.client.constants*), 12
- ProviderConfig (class in *centralauth.provider.apps*), 16
- R**
- ready() (*centralauth.client.apps.ClientConfig* method), 12
- refresh_token_set (*centralauth.provider.models.Application* attribute), 18
- register_perms() (in module *centralauth.client.services*), 13
- repr (*centralauth.provider.models.ApplicationPermission* attribute), 19
- S**
- save() (*centralauth.provider.models.Application* method), 16
- save_related() (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* method), 14
- save_token() (in module *centralauth.client.services*), 13
- search_fields (*centralauth.provider.admin.ApplicationPermissionGroup* attribute), 14
- search_fields (*centralauth.provider.admin.ApplicationUserAdmin* attribute), 15
- serialize_perm() (in module *centralauth.client.services*), 13
- sync_user() (in module *centralauth.client.services*), 13
- U**
- update_user() (in module *centralauth.client.services*), 13
- user (*centralauth.provider.models.Application* attribute), 18
- user (*centralauth.provider.models.ApplicationUser* attribute), 21
- user_details() (in module *centralauth.client.services*), 13
- user_id (*centralauth.provider.models.ApplicationUser* attribute), 22
- UserEndpoint (class in *centralauth.provider.views*), 22
- W**
- widgets (*centralauth.provider.admin.ApplicationPermissionGroupAdmin* attribute), 14